Video intercom with network interface

Marcin Słoniewski

January 26, 2016

Intermediate project
Instructor: dr inż. Witold Paluszyński [1]

The goal of this project was to develop intercom based on Raspberry Pi, which could recognize faces using camera and show results on the PC within the same network.
Results: Working intercom with face detection, without recognition of pre-learned types of faces.

[1] Wydział Elektroniki, Politechnika Wrocławska, ul. Z. Janiszewskiego 11/17, 50-372 Wrocław

1 Introduction

The project goal was to develop platform to recognize people waiting at the door in front of house and send information to PC located in the same network. The system would be triggered by proximity sensor, which detects if something is present in front of him. When something is detected system would take photo and use image processing algorithm to locate faces and check if they are in database of known faces. The result would be sent to the computer with graphical program to show it to the user.
Functionalities:

- Taking photos with dedicated camera
- Detecting faces in a picture
- Checking if detected face is in database
- Graphical program on another computer to communicate with station within LAN
- Detecting people's presence with proximity sensor

2 Hardware set-up

Components, which were used in project:

- Raspberry Pi B+
- Proximity sensor E18-D80NK 5-80 cm
- Dedicated Camera HD B for Raspberry Pi
- Access point with Ethernet port

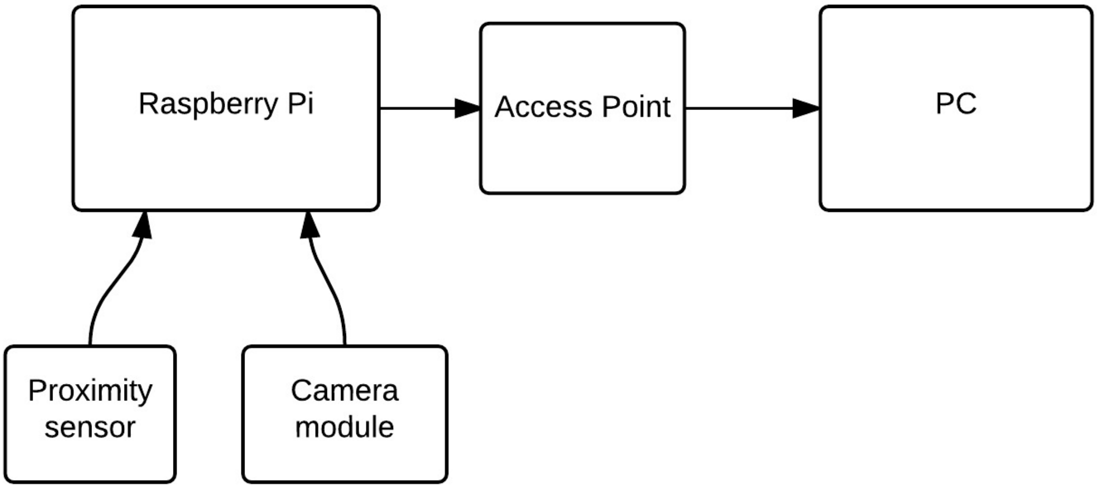

Figure 1: Scheme how components were connected together.

3 Software

3.1 Software on Raspberry Pi

On Raspberry Pi was installed operating system Raspbian. The whole program was written in Python 2.7. For communication with PC I used raw UDP socket to inform about new photo to download. For sending photo I used ProFTPD, which sets up FTP server, which runs independently from main program and services request for download from PC.
Face detection algorithm was implemented with OpenCV library - open-source software focused on real-time computer vision. The was no pre-compiled library, so I was forced to download sources files and compile them on Raspberry Pi. In package there was default Haar classifier for face detection which I used.

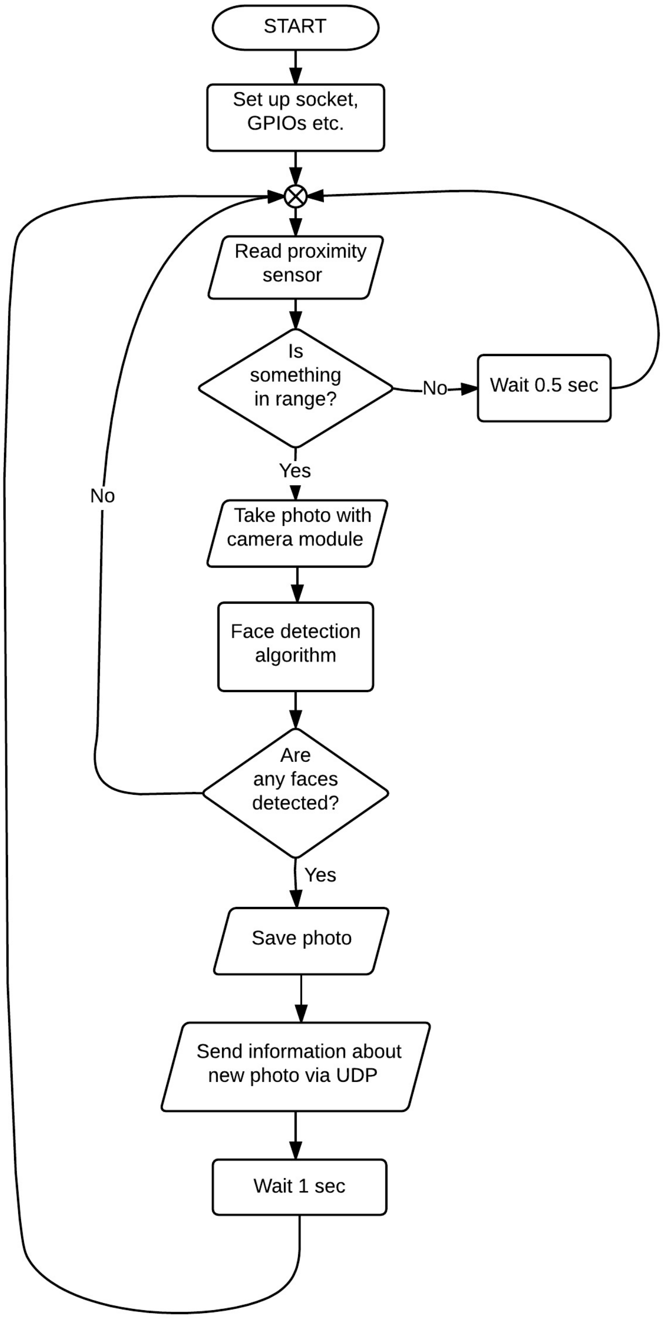

Figure 2: Program for Raspberry Pi - Block diagram.

3.2 Software on PC

The program for PC was written in Python 2.7 as well. For handling graphics in window i used Tkinter - standard GUI package for Python. In the beginning program shows standard message in the window ”Waiting for image...”, until intercom module send over UDP information about new image with face captured. Then the program log in into FTP server on intercom module and downloads newest image, which will be shown i window. FTP handler was implemented with python module ftplib.py.

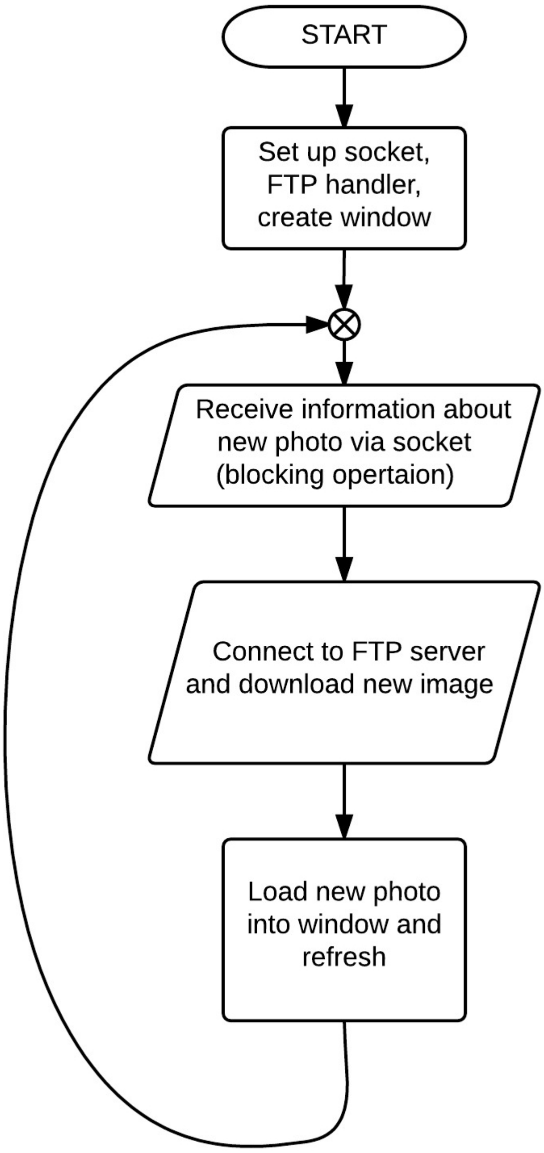
START
Set up socket, FTP handler, create window
Receive information about new photo via socket (blocking opertaion)
Connect to FTP server and download new image
Load new photo into window and refresh

Figure 3: Program for PC - Block diagram.

4 Conclusions

During this project I achieved only minimal version, that is: detection of presence with proximity sensor, taking photo and detect faces in it, then send to PC and idiomatically show it on the screen.
I was able to learn about:

- Basic ideas about Haar Classifier
- Capabilities of OpenCV library
- Using dedicated camera module with Raspberry Pi
- Setting up and using FTP server

5 Used software with references

- OpenCV (BSD license) - http://opencv.org/
- ProFTPD (GPL) - http://www.proftpd.org/
- Tkinter (BSD license)
  - http://www.tcl.tk/
  - http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/index.html
- RPi.GPIO (GNU Lesser General Public License)- https://pypi.python.org/pypi/RPi.GPIO
- Raspbian (GPLv2) - https://www.raspbian.org/